IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

Analysis of Software Quality Models for ERP Software Use in University in Kenya

Dickson Osore Waliaro[1], Kelvin Omieno[2], Jasper Ondulo[3]

Computing & Informatics, MMUST, Kakamega, Kenya[1,3]

Computing & Informatics, KAFUCO, Kaimosi, Kenya[2]

Abstract: Software quality models act as a guide to improve the use and activities of existing software systems. Such activities can be among others include improved functionality and usability of such systems in learning organizations. Quality models play a big role in ensuring the above objectives are met. Several quality models have been used to facilitate coming up with different system software for use in higher learning institutions. Different scholars have come up different software quality models to help measure the quality of software products. In my research, we are going to discuss the different quality models compare the quality models with each other in relation to ERP use in universities in Kenya, show their criticism in relation to functionality and usability. In addition, a framework containing steps is proposed by the author. Some recommendations are also framed hereby in the following research paper.

Keywords: Boehm Model, Functionality, FURPS Model, McCall Model, Software Quality Models

I. INTRODUCTION

Software functionality is one of the most important components of any software. Functional suitability is the degree to which the software product provides functions that meet stated and implied needs when the software is used under specified conditions (ISO/IEC, 2011a).many software- dependent systems in institutions have failed to meet the needs of their users. The later failures in their uses and functionality indicate a big gap between the users' needs and the documented requirements against which the system and its sub sets are verified (Mika, n.d.). For these software to carry out their intended needs, their quality has to be put into consideration.

Quality models are the tools for focusing software development efforts (James, 205AD). They are used to identify program modules that are likely to be defective (Taghi, 2009). They also help in using the resources effectively. a quality model is "the set of characteristics, and the relationships between them that provides the basis for specifying quality requirements and evaluation". The models to evaluate the quality of software have been constructed defining the fundamental factors (also called characteristics), and within each of them the sub factors (or sub characteristics). Metrics are assigned to each sub factor for the real evaluation (ISO/IEC, 2001a)(Jose' et Al, 2014). In this paper, I will discuss the following quality models:

Boehm's Quality Model
FURPS's Quality Model
McCall's Quality Model

In addition, we will focus on the comparison between the quality models, their weaknesses which will lead to their critism and also look at their differences.

McCall's Quality Model

McCall's Quality Model (also known as the General Electric's Model of 1977) is one of the most known quality models in the software engineering literature. McCall software quality model is aimed towards system developers and system development process (T. et Al, 2011). McCall's model is to bridge the gap between users and developers by focusing on a number of software quality factors that indicate the views of both users and developers. McCall's model looks at three major aspects, product revision, product transition and product operations (McCall et Al, 1977).

The model consists of 11 quality factors to describe the external view of the software (users' view); 23 quality criteria to describe the internal view of the software (developer's view); and a set of metrics that are used for quality evaluation. The category product revision consists of maintainability, flexibility, and testability quality attributes. The product transition category consists of portability, reusability, interoperability quality attributes (T. et Al, 2011), (Jawad, 2013). The product operation category consists of a set of quality attributes that includes correctness,

reliability, usability, integrity, and efficiency. The fundamental idea of this model is assessing the relationship among external quality factors and product quality criteria. A major contribution of this model is the relationship between quality characteristics and metrics (Jawad, 2013).

Criticism of the McCall's Quality model (1977)

However, there are criticisms such as not all metrics are objectives (Al-Qutaish, 2010) and the functionality of software product is not considered in this model (Behkamal et Al, 2009). Furthermore, the model completely leaves out the functional suitability in organizations including the education institutions (F. et Al, 2012). Architectural integrity is not covered in the model. Moreover, none of the factors or quality criteria in the model is related to architectural integrity with respect to the understanding and coherence to the architectural decisions (Jamwal, 2010b).This model is proposed for general application systems, and thus the domain specific attributes are not explicitly addressed in the scope of the model. Furthermore, another drawback of the McCall model is the accuracy in the measurement of quality, as it is based on responses of Yes or No. Furthermore, the model does not consider the functionality so that the user's vision is diminished.

Boehm Model (1978)

Boehm introduced a model for evaluating the quality of software both automatically and quantitatively (Al-Qutaish, 2010). It presents a hierarchical structure similar to McCall consisting of High-Level, Intermediate-Level and Low-Level Characteristics. Each of these characteristics contributes to the total quality of software product. This model takes into account some considerations of software product with respect to the utility of the program. Boehm also extended characteristics to the McCall model by emphasizing the Maintainability factor of a software product, which is one of the advantages of this model. However, it does not suggest any approach to measure its quality characteristics (McCall et Al, 1977). The primitive characteristics can be used to provide the foundation for defining quality metrics, this use is one of the most important goals established by Boehm when he constructed his quality model. One or more metrics are supposed to measure a given primitive characteristic. Boehm defined the 'metric' as "a measure of extent or degree to which a product possesses and exhibits a certain (quality) characteristic."

Criticism of the Boehm Model (1978)

Just like McCall, Boehm model ignores the functionality aspects of the software which is not mentioned anywhere within the model (F. et Al, 2012). Neither in the Boehm quality model is all the software evolvability sub characteristics explicitly addressed. Analyzability is partially addressed through the characteristic understandability, which describes that the purpose of the code is clear to the inspector (Jamwal, 2010b). However, none of the factors or measurable properties describes the capability to analyze the impact at the software architecture level due to a change stimulus. In this model also architectural integrity is not covered in the model.

FURPS Model

Robert Grady and Hewlett Packard proposed the FURPS model that decomposes characteristics into 2 categories of requirement: Functional Requirements and Non- Functional Requirements (Maryoly et Al, 2002). Functional requirements (F) are defined by input and expected output while non-functional requirements (URPS) consist of usability, reliability, performance and supportability. FURPS takes into account the five characteristics that make up its name: Functionality, Usability, Reliability, Performance, and Supportability (Maryoly et Al, 2002).

Criticism of the FURPS Model

It is important to note that domain specific attributes and software product portability were not addressed in this model (Grady, 1992). The model does not consider the subsets of functionality such as suitability, accurateness, interoperability, security and compliance (F. et Al, 2012). None of the characteristics or sub characteristics in the model is related to architectural integrity with respect to the understanding and coherence to the architectural decisions. Moreover, one disadvantage of this model is that it fails to take account of the software portability (Maryoly et Al, 2002). Domain-specific attributes are not addressed either in the model (Jamwal, 2010b).

Methodology

This section discusses the chosen research methodology.

The method selected used questionnaires to measure the impact of lifecycles on the factors that influence the use of ERP software use in universities in Kenya. The resultant data was used to drive a model selection framework. The framework suggested how suitable a lifecycle model is for a given project. The model recommendations was examined and discussed using a set of case studies.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 10, October 2019

Questionnaires provide a structured approach to gathering data; and closed questions provide a limited list of responses, ensuring easy transcription for processing. This makes closed questions suitable for gathering lifecycle impact data, while open questions were used to solicit and gathered from free-text entry boxes.

Model choice is often made automatically and without consciously considering alternative models. This suggests volunteers may need time to gather their thoughts before answering model selection questions. Questionnaires are ideal in this situation, since there is no set time limit for completing them.

II. ANALYSIS / COMPARISON

In this research paper, we have studied different types of software models like Boehm's quality model, FURPS's quality model, and Mc Call's quality model. The limitations of the models after the analysis are summarized as shown below;

MODEL	LIMITATIONS
McCall Quality Model	✓ Metrics are not objective. ✓ Software functionality is not considered. ✓ Functional suitability in organizations (including HE institutions) not covered. ✓ No quality architectural integrity ✓ Proposed for general applications only ✓ No accuracy in measurement of quality
Boehm Quality Model	✓ Software functionality is not covered. ✓ No quality architectural integrity. ✓ No software evolvability ✓ No architectural integrity
FURPS Quality Model	✓ Software portability not addressed. ✓ Software functionality not considered. ✓ Subsets of functionality are ignored ✓ No architectural integrity

Proposal

1. Define the universities needs and goals for software quality. It is essential to understand users priorities, preferences, regulations among others.
2. Identify which quality elements are most important to the Enterprise Resource Planning.
3. Develop details and examples to explain software quality factors which are most important to the institution.
4. Build the quality factors and the quality model into development and test methodologies.
5. After the procedures above, you have organized a software development, test and quality process which systematically addresses the software quality elements which match the institutions strategic goals. However, technology changes constantly hence need to communicate with the staff and ERP users to ensure agreement on goals and priorities.

Recommendations

1. The stakeholders of ERP need to identify a small set of agreed-upon, high-level quality attributes, and in a top-down fashion decompose each attribute into a set of subordinate attributes.
2. Distinguish between internal and external metrics
3. Identify type of users for each high-level quality attributes.
4. Put the pieces together, constructing the new models that implement ideas from international standards.

III. CONCLUSION

We have studied different types of software quality models in software engineering. Each of these quality models consists of number of characteristics. Selecting which one of the quality models to use is a real challenge. In this paper, we have discussed and compared the following quality models:

1. McCall's Quality Model.
2. Boehm's Quality Model
3. FURPS Quality Model

Different analysis have been made and discussed based on the limitations of the models in terms of use in the ERP. They provide a wide base on the implementation of the same in learning institutions.

REFERENCES

[1]. ISO/IEC. (2011a). Software Engineering-Systems and Software Quality Requirements and Evaluation. ISO/IEC.
[2]. Mika, S. &. (n.d.). Quality Requirements for Software Dependent Safety Critical Systems History, Current Status, and Future Needs.
[3]. James, P. &. (205AD). A Probabilistic Model for Predicting Software Development Efforts. IEEE, 31(7).
[4]. Taghi, K. &. (2009). An Empirical Investigation of Filter Attributes Selection Techniques for Software Quality Classification. 10–12.
[5]. Al, Jose' et. (2014). A Review of Software Quality Models for the Evaluation of Software Products. IJSEA, 5(6).
[6]. Al, T. et. (2011). A Systematic Study of SOftware Quality Models. IJSEA, 2(4).
[7]. Al, McCall et. (1977). Factors in Software Quality. 1.
[8]. Jawad, K. &. (2013). Quality Model Based on Cots Quality Attributes. IJSEA, 4.
[9]. Al-Qutaish. (2010). Quality Models in Software Engineering Literature. 6(3), 166–175.
[10]. Al, Behkamal et. (2009). Customizing ISO 9126 Quality Model for Evaluation of B2B Applications. JIST, 51(3).
[11]. Al, F. et. (2012). Evaluating the Quality of Software in e-Book Using the ISO 9126 Model. IJCA, 5(2).
[12]. Jamwal. (2010b). Analysis of Software Quality Models for Organizations. IJLTC, 1(2).
[13]. Al, Maryoly et. (2002). A Systematic Quality Model for Evaluating Software Products.
[14]. Grady. (1992). Practical Software Metrics for Project Management and Process Improvement.